

I/We Claim:

1. An apparatus, comprising:
a message dispatcher that routes and dispatches messages, wherein each message is routed based on an arbitrary portion of the message's contents; and
an interface through which application programs communicate with the message dispatcher to define the arbitrary portion of the message's contents.
2. The apparatus of claim 1, wherein the message dispatcher comprises a transport independent message dispatcher, and the message dispatcher communicates using a transport independent protocol.
3. The apparatus of claim 1, wherein the message dispatcher routes a first network message based on a first attribute of said first network message, and routes a second network message based on a second attribute, different from said first attribute, of said second network message.
4. The apparatus of claim 1 wherein the message dispatcher routes a first network message, addressed to a recipient from a first sender, to a first server, and
wherein the message dispatcher routes a second network message, addressed to the recipient from a second sender, to a second server.
5. The apparatus of claim 1, wherein the message dispatcher routes messages using a virtual network protocol above a transport layer protocol.
6. The apparatus of claim 1, further comprising a transport adapter to convert messages between the transport layer protocol and the virtual network protocol.
7. The apparatus of claim 1, wherein the arbitrary portion of the message's contents comprises an application level header.

8. A data processing apparatus, comprising:
a message dispatcher module;
a transport adapter for interfacing the message dispatcher to a transport protocol;
an interface through which application programs communicate with the message dispatcher;
stored rules instructing the message dispatcher to route a first network message based on a first attribute of said first network message, and route a second network message based on a second attribute, different from said first attribute, of said second network message, wherein the first and second attributes are selected from a set of headers and data contained in each network message.

9. The data processing apparatus of claim 8, wherein the first attribute comprises an application created header.

10. The data processing apparatus of claim 8, wherein each message rule is stored in a message handler.

11. The data processing apparatus of claim 10, comprising a first message handler that, upon the occurrence of a predetermined condition, alters a second message handler.

12. The data processing apparatus of claim 10, comprising a first message handler that, upon the occurrence of a predetermined condition, alters the first message.

13. The data processing apparatus of claim 11, wherein the predetermined condition comprises a nonoccurrence of an event.

14. The data processing apparatus of claim 13, wherein the message dispatcher module comprises computer executable instructions that, when executed, cause the data processing apparatus to perform the steps of:

- (i) polling a second apparatus in first predetermined intervals; and

(ii) receiving poll responses from the second apparatus;
and wherein the predetermined condition comprises the nonoccurrence of step (ii) for a predetermine amount of time.

15. The data processing apparatus of claim 14, wherein when the predetermined condition is met, the message dispatcher alters the second message handler to redirect messages, that were originally addressed to the second apparatus, to a third apparatus.

16. The data processing apparatus of claim 15, wherein the computer executable instructions further cause the data processing apparatus to perform the step of sending routing information to a second message dispatcher, indicating the change of routing information corresponding to the second and third apparatus.

17. A method for routing network messages, comprising the steps of:
(i) routing a first network message based on a first attribute of the first network message;
(ii) routing a second network message based on a second attribute, different from said first attribute, of said second network message;
wherein the first and second attributes may be any field selected from a set of headers and data of each network message.

18. The method of claim 17, further comprising the steps of:
(iii) receiving instructions comprising a message field and a field condition;
(iv) modifying a message handler based on the received instructions.

19. The method of claim 18, wherein, in step (iii), the instructions are received from a network application program.

20. The method of claim 18, wherein, in step (iii), the instructions are based on user-input.

21. The method of claim 17, wherein, in steps (i) and (ii), each message is output to a transport adapter that converts the message from a virtual network protocol to a transport protocol.

22. The method of claim 17, wherein, in step (i), the first attribute comprises an application created header.

23. The method of claim 17, further comprising the step of storing routing instructions in message handlers, and
wherein steps (i) and (ii) are performed based on stored message handlers.

24. The method of claim 23, further comprising the step of altering a first message handler when a predetermined condition occurs.

25. The method of claim 23, further comprising the step of altering a network message when the message meets a predetermined condition stored in a message handler.

26. The method of claim 24, wherein the predetermined condition comprises a nonoccurrence of an event.

27. The method of claim 17, further comprising the steps of:
(iii) polling a first data processing device in predetermined intervals;
(iv) receiving poll responses from the first data processing device; and
(v) when step (iv) has not occurred for a predetermined amount of time, altering a message handler to direct messages originally addressed to the first data processing device, to a second data processing device.

28. The method of claim 27, further comprising the step of sending routing information to a message dispatcher, indicating the change of routing information corresponding to the first and second data processing devices.

29. A network router comprising computer executable instructions that, when executed by the router, perform steps of:

- (i) storing routing information received from a network application, wherein the routing information comprises a message field, a field condition, and a routing instruction;
- (ii) receiving a network message;
- (iii) processing the network message by comparing the network message to the stored routing information;
- (iv) when the received message's message field meets the field condition, performing the routing instruction.

30. The network router of step 29, wherein, in step (iv), the routing instruction comprises altering the message.

31. The network router of step 29, wherein, in step (iv), the routing instruction comprises routing the message based on application level header.

32. A computer network, comprising:
a plurality of computers, each comprising:

at least one transport adapter that converts messages between a transport layer protocol and a network protocol; and

a message dispatcher that routes and dispatches messages based on an arbitrary portion of the message's contents, and wherein the message dispatcher in each computer routes messages in the virtual network protocol over the transport layer protocol using the at least one transport adapter.

33. The computer network of claim 32, wherein a first message dispatcher in a first computer is configurable for use with a new transport protocol by adding a new transport adapter that converts messages between the new transport layer protocol and the network protocol, without requiring a network application to be reconfigured for use with the new transport protocol.

34. A virtual network, comprising at least one virtualized component inserted between layer 7 and layer 6 of an OSI protocol stack.

35. The virtual network of claim 34, wherein the at least one virtualized component comprises a virtual network message dispatcher.

36. The virtual network of claim 34, wherein the at least one virtualized component comprises a synchronization module.

37. The virtual network of claim 34, wherein the at least one virtualized component comprises an eventing module.

38. The virtual network of claim 34, wherein the at least one virtualized component comprises a names module.

39. The virtual network of claim 34, wherein the at least one virtualized component comprises a groups module.

40. The virtual network of claim 34, wherein the at least one virtualized component comprises an addressing module.

41. The virtual network of claim 34, wherein the at least one virtualized component comprises a security module.

42. The virtual network of claim 34, wherein the at least one virtualized component comprises an administrative module.

42. The virtual network of claim 34, wherein the at least one virtualized component comprises an administrative module.